Information and Software Technology 56 (2014) 991–1011

Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof

A noun-based approach to feature location using time-aware term-weighting

Sima Zamani [a,*], Sai Peck Lee [a], Ramin Shokripour [a], John Anvik [b]

[a] *Faculty of Computer Science & Information Technology, University of Malaya, Kuala Lumpur, Malaysia*
[b] *Department of Computer Science, Central Washington University, Ellensburg, WA, USA*

ARTICLE INFO

Article history:
Received 19 August 2013
Received in revised form 17 March 2014
Accepted 18 March 2014
Available online 26 March 2014

Keywords:
Feature location
Software change request
Time-metadata
Term-weighting
Noun usage

ABSTRACT

Context: Feature location aims to identify the source code location corresponding to the implementation of a software feature. Many existing feature location methods apply text retrieval to determine the relevancy of the features to the text data extracted from the software repositories. One of the preprocessing activities in text retrieval is term-weighting, which is used to adjust the importance of a term within a document or corpus. Common term-weighting techniques may not be optimal to deal with text data from software repositories due to the origin of term-weighting techniques from a natural language context.
Objective: This paper describes how the consideration of when the terms were used in the repositories, under the condition of weighting only the noun terms, can improve a feature location approach.
Method: We propose a feature location approach using a new term-weighting technique that takes into account how recently a term has been used in the repositories. In this approach, only the noun terms are weighted to reduce the dataset volume and avoid dealing with dimensionality reduction.
Results: An empirical evaluation of the approach on four open-source projects reveals improvements to the accuracy, effectiveness and performance up to 50%, 17%, and 13%, respectively, when compared to the commonly-used Vector Space Model approach. The comparison of the proposed term-weighting technique with the Term Frequency-Inverse Document Frequency technique shows accuracy, effectiveness, and performance improvements as much as 15%, 10%, and 40%, respectively. The investigation of using only noun terms, instead of using all terms, in the proposed approach also indicates improvements up to 28%, 21%, and 58% on accuracy, effectiveness, and performance, respectively.
Conclusion: In general, the use of time in the weighting of terms, along with the use of only the noun terms, makes significant improvements to a feature location approach that relies on textual information.

1. Introduction

During software evolution, the existing source code of a project undergoes incremental modifications in order to satisfy software change requests [1–3]. A change request may result in adding a new software feature, removing a bug or defect, or improving existing software functionality [4]. More effective support for change requests is needed to obtain a sustainable, high-quality evolution of a software system. One of the key issues in addressing a change request is finding relevant locations in the source code of the project, such as files, classes, or methods, requiring modification to address the change request [2,5]. Performing this process manually in a large-scale software project is challenging and time-consuming.

Feature location [6,7] is a well-known technique used by software developers to address this challenge and it has become one of the most frequent program comprehension activities. Feature location aims to identify the initial location in the source code that is pertinent to a change request. It is also part of other software evolution tasks, such as the recovery of traceability links between software repositories, and the retrieval of software components for reuse [8].

A recent survey of feature location literature [9] found that more than 51% of the published literature in this research area is based, at least in part, on text retrieval. The primary text retrieval methods reported in this literature are: Pattern Matching (PM) [10,11], Information Retrieval (IR) [12,13] and Natural Language Processing (NLP) [14,15]. The use of these methods is based on the assumption that identifiers, comments, and other text data found in software repositories contain domain knowledge that can be used for locating software features [16]. However, these

[*] Corresponding author. Tel.: +60 147367656.
E-mail addresses: s.zamani@siswa.um.edu.my, sima.zamani@gmail.com (S. Zamani).

http://dx.doi.org/10.1016/j.infsof.2014.03.007
0950-5849/© 2014 Elsevier B.V. All rights reserved.

methods originate from a natural language context such as the summarization of newspaper articles which are less structured than the text documents found in software repositories [17]. Furthermore, unlike the typical context in which these methods are applied, text documents in software repositories have a corresponding set of metadata [18]. In other words, the analysis of the textual data found in software repositories requires different techniques and methods than for those found in other text analysis domains.

As previously mentioned, text documents in software repositories are associated with a set of metadata. This metadata includes such items as developer identifiers, time stamps, and commit comments. This metadata associates answers of who, when, and why with the data in a software repository [19]. One important piece of metadata is the 'when', or the time at which the data was created or modified. According to the feature location literature, metadata has only been used by Sisman and Kak [20] for the weighting of files in a probabilistic IR model. As demonstrated in their work, the use of time-metadata can enhance feature location approaches and improve their results. In other words, the linking of the text data in a software repository with the associated time-metadata indicates that the importance of text varies over the different periods of the project's life. In this paper, we propose a new feature location approach that includes weighting and ranking the source code locations based on both the textual similarity with a change request, and the use of time-metadata. The consideration of time-metadata in feature location is based on the following assumption. For a new change request, the source code entities with the highest textual similarity, and have also been most recently modified, lead to the most relevant source code locations. This assumption is based on two principles:

- Defect localization: It is known that the most recent modifications to a project are most likely the cause of future bugs or defects [21,22]. By considering recent modifications, this may lead to finding relevant locations that are the cause of a new change request [20].
- Software evolution: Each software project has different goals and requirements in different periods of the project's life [23]. For a given change request, the requested modification to the source code in the same time period of project's life will likely have the same goals or requirements. This principle bears further elaboration.

In every period of a project's life cycle, the terms that are used across the different repositories, such as source code version and issue tracking, are consistent with the requirements of the project during that specific time period [23]. Change requests occurring in a different time period of the project's life may have different goals. For instance, the change requests which are reported in the initial period of the project life are usually focused on the fundamental requirements of the project. A common way of addressing this type of change request is to create new file(s) or make extensive modifications to existing files. Consequently, it is common to have a large number of modified files resulting from this set of change requests, and the changes that are made are correspondingly extensive.

This claim is supported by the systems used in this study. For example, in the first working day of the JDT[1] project, around 490 files were modified in 490 commits to the project's source code repository. Similarly, in the first day of the AspectJ[2] project, 87 files were modified in 2873 commits. The number of modified files and the extent of modifications gradually decreases over time and becomes stable. The time needed to reach this stability depends on the age of the project. Therefore, the time stamp of when a term was used in the project can play a significant role in determining the degree of relevancy of the term with a change request. This observation has motivated a new term-weighting technique that considers the value of the text over time.

In addition, traceability research conducted by Capobianco et al. [24,25] showed that using only the noun terms of the text greatly improves the accuracy of IR-based traceability recovery methods. Inspired by this result, the use of only noun terms in a bug assignment process was investigated and found to improve the accuracy of the developer recommendation [26].

This paper presents a feature location approach that uses only noun terms, called Noun-Based Feature Location (NBFL), and a time-aware weighting technique, called Time-Aware Term-Weighting (TATW). Text data is extracted from two software repositories of project, namely, Version Control System (VCS) and Issue Tracking System (ITS). The VCS is a software repository that manages the changes of source code and its relevant documents and the ITS is a software repository that tracks reported software change requests.

This paper also presents an empirical evaluation of the proposed approach. The evaluation was conducted on four open-source projects of varying sizes and development speeds. The result of using NBFL is compared to that of using the Vector Space Model (VSM) with the Term Frequency-Inverse Document Frequency (TF-IDF) technique [27]. VSM is a high-performing IR model in feature location that is used with common term-weighting techniques, and TF-IDF was chosen to assess the proposed term-weighting technique. The evaluation setup is similar to that used in the work of Rao and Kak [28] and Zhou et al. [29].

The evaluation showed a number of results. First, there are benefits to using only the noun terms from the text data. These benefits include:

- An independence of the approach from dimensionality reduction methods, which is one of the challenges with IR methods [30].
- A reduced amount of noise in the extracted entities from the data-sources [31], thereby enhancing the effectiveness of improvements made by other means.
- The use of only nouns provides enough information to make a feature location decision for a given change request (Section 4).

Second, the evaluation of the approach found that the NBFL approach using TATW outperformed the VSM approach in accuracy, effectiveness, and performance by as much as 50%, 17% and 13%, respectively. The evaluation also found that the TATW technique outperformed TF-IDF in accuracy, effectiveness, and performance by as much as 15%, 10% and 40%, respectively. In addition, an evaluation of the impact of using only noun terms in the proposed approach indicated an improvement on the accuracy, effectiveness, and performance of a feature location approach by up to 28%, 21%, and 58%, respectively. In summary, these results show that the use of time-metadata in a noun-based feature location approach is an improvement over the standard feature location approach using VSM and TF-IDF.

The remaining sections of this paper are organized as follows. In Section 2, the proposed feature location approach (NBFL) is described in detail. The setup for the empirical evaluation is presented in Section 3 and the evaluation results are given and discussed in Section 4. In Section 5, some of the threats to the validity of this study are outlined. An overview of the related research in feature location using text resources is presented in

[1] http://www.eclipse.org/jdt/.
[2] http://www.eclipse.org/aspectj/.